

# Neural Stochastic Contraction Metrics for Learning-Based Control and Estimation

Hiroyasu Tsukamoto<sup>ID</sup>, *Graduate Student Member, IEEE*, Soon-Jo Chung<sup>ID</sup>, *Senior Member, IEEE*,  
and Jean-Jacques E. Slotine<sup>ID</sup>

**Abstract**—We present Neural Stochastic Contraction Metrics (NSCM), a new design framework for provably-stable learning-based control and estimation for a class of stochastic nonlinear systems. It uses a spectrally-normalized deep neural network to construct a contraction metric and its differential Lyapunov function, sampled via simplified convex optimization in the stochastic setting. Spectral normalization constrains the state-derivatives of the metric to be Lipschitz continuous, thereby ensuring exponential boundedness of the mean squared distance of system trajectories under stochastic disturbances. The trained NSCM model allows autonomous systems to approximate optimal stable control and estimation policies in real-time, and outperforms existing nonlinear control and estimation techniques including the state-dependent Riccati equation, iterative LQR, EKF, and the deterministic NCM, as shown in simulation results.

**Index Terms**—Machine learning, stochastic optimal control, observers for nonlinear systems.

## I. INTRODUCTION

THE KEY challenge for control and estimation of autonomous aerospace and robotic systems is how to ensure optimality and stability. In general, their motions are expressed as nonlinear systems with unbounded stochastic disturbances, the time evolution of which is governed by Itô stochastic differential equations [1]. As their onboard computational power is often limited, it is desirable to execute control and estimation policies computationally as cheaply as possible.

In this letter, we present a Neural Stochastic Contraction Metric (NSCM) based robust control and estimation framework outlined in Fig. 1. It uses a spectrally-normalized neural network as a model for an optimal contraction metric

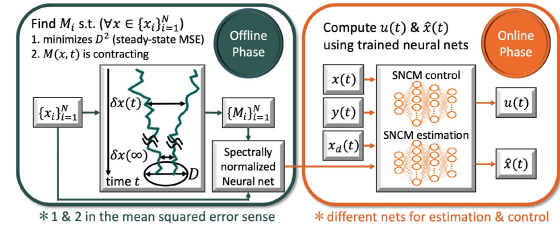


Fig. 1. Illustration of NSCM ( $M(x, t)$ : optimal contraction metric;  $x_i$  and  $M_i$ : sampled states and contraction metrics;  $y(t)$ : measurements;  $x(t)$ ,  $x_d(t)$ , and  $\hat{x}(t)$ : actual, target, and estimated trajectories, respectively.

(differential Lyapunov function), the existence of which guarantees exponential boundedness of the mean squared distance between two system trajectories perturbed by stochastic disturbances. Unlike the Neural Contraction Metric (NCM) [2], where we proposed a learning-based construction of optimal contraction metrics for control and estimation of nonlinear systems with bounded disturbances, stochastic contraction theory [3]–[5] guarantees stability and optimality in the mean squared error sense for unbounded stochastic disturbances via convex optimization. Spectral Normalization (SN) [6] is introduced in the NSCM training, in order to validate a major assumption in stochastic contraction that the first state-derivatives of the metric are Lipschitz. We also extend the State-Dependent-Coefficient (SDC) technique [7] further to include a target trajectory in control and estimation, for the sake of global exponential stability of unperturbed systems.

In the offline phase, we sample contraction metrics by solving convex optimization to minimize an upper bound of the steady-state mean squared distance of stochastically perturbed system trajectories (see Fig. 1). Other convex objectives such as control effort could be used depending on the application of interest. We call this method the modified CV-STEM (mCV-STEM), which differs from the original CV-STEM [8] in the following points: 1) a simpler stochastic contraction condition with an affine objective function both in control and estimation, due to the Lipschitz condition on the first derivatives of the metrics; 2) generalized SDC parameterization, i.e.,  $A$  s.t.  $A(x, x_d, t)(x - x_d) = f(x, t) + B(x, t)u_d - f(x_d, t) - B(x_d, t)u_d$  instead of  $A(x, t)x = f(x, t)$ , for systems  $\dot{x} = f(x, t) + B(x, t)u$ , which results in global exponential stability of unperturbed systems even with a target trajectory,  $x_d$  for control and  $x$  for estimation; and 3) optimality in the contraction rate  $\alpha$  and disturbance attenuation parameter  $\epsilon$ . The second point is in fact general, since  $A$  can always be selected based on the

Manuscript received September 14, 2020; revised November 18, 2020; accepted December 7, 2020. Date of publication December 22, 2020; date of current version January 13, 2021. This work was supported in part by the Raytheon Company. Recommended by Senior Editor G. Cherubini. (Corresponding author: Hiroyasu Tsukamoto.)

Hiroyasu Tsukamoto and Soon-Jo Chung are with the Graduate Aerospace Laboratories, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: htsukamoto@caltech.edu; sjchung@caltech.edu).

Jean-Jacques E. Slotine is with the Nonlinear Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jjs@mit.edu).

Digital Object Identifier 10.1109/LCSYS.2020.3046529

2475-1456 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

line integral of the Jacobian of  $f(x, t) + B(x, t)u_d$ , a property which can also be applied to the deterministic NCM setting of [2]. We then train a neural network with the sampled metrics subject to the aforementioned Lipschitz constraint using the SN technique. Note that reference-independent integral forms of control laws [9]–[13] could be considered by changing how we sample the metrics in this phase. Our contraction-based formulation enables larger contracting systems to be built recursively by exploiting combination properties [14], as in systems with hierarchical combinations (e.g., output feedback or negative feedback), or to consider systems with time-delayed communications [15].

In the online phase, the trained NSCM models are exploited to approximate the optimal control and estimation policies, which only require one neural network evaluation at each time step as shown in Fig. 1. The benefits of this framework are demonstrated in the rocket state estimation and control problem, by comparing it with the State-Dependent Riccati Equation (SDRE) method [5], [7], Iterative LQR (ILQR) [16], [17], EKF, NCM [2], and mCV-STEM.

*Related Work:* Contraction theory [14] is an analytical tool for studying the differential dynamics of a nonlinear system under a contraction metric, whose existence leads to a necessary and sufficient characterization of its exponential incremental stability. The theoretical foundation of this letter rests on its extension to stability analysis of stochastic nonlinear systems [3]–[5]. The major difficulty in applying it in practice is the lack of general analytical schemes to obtain a suitable stochastic contraction metric for nonlinear systems written as Itô stochastic differential equations [1].

For deterministic systems, there are several learning-based techniques for designing real-time computable optimal Lyapunov functions/contraction metrics. These include [2], [18], [19], where neural networks are used to represent the optimal solutions to the problem of obtaining a Lyapunov function. This letter improves our deterministic NCM [2], as the NSCM explicitly considers the case of stochastic nonlinear systems, where deterministic control and estimation policies could fail due to additional derivative terms in the differential of the contraction metric under stochastic perturbation.

The CV-STEM [8] is derived to construct a contraction metric accounting for the stochasticity in dynamical processes. It is designed to minimize the upper bound of the steady-state mean squared tracking error of stochastic nonlinear systems, assuming that the first and second derivatives of the metric with respect to its state are bounded. In this letter, we only assume that the first derivatives are Lipschitz continuous, thereby enabling the use of spectrally-normalized neural networks [6]. This also significantly reduces the computational burden in solving the CV-STEM optimization problems, allowing autonomous agents to perform both optimal control and estimation tasks in real-time.

## II. PRELIMINARIES

We use  $\|x\|$  and  $\|A\|$  for the Euclidean and induced 2-norm,  $I$  for the identity matrix,  $E[\cdot]$  for the expected value,  $\text{sym}(A) = (A + A^T)/2$ , and  $A \succ 0$ ,  $A \geq 0$ ,  $A \prec 0$ , and  $A \leq 0$  for positive definite, positive semi-definite, negative definite, and negative semi-definite matrices, respectively. Also,  $f_x$  is the partial derivative of  $f(x, t)$  respect to the state  $x$ , and  $M_{x_i}$  is of  $M(x, t)$  with respect to the  $i$ th element of  $x$ ,  $M_{x_i x_j}$  is of  $M(x, t)$  with respect to the  $i$ th and  $j$ th elements of  $x$ .

## A. Neural Network and Spectral Normalization

A neural network is a mathematical model for representing training samples  $\{(x_i, y_i)\}_{i=1}^N$  of  $y = \phi(x)$  by optimally tuning its hyperparameters  $W_\ell$ , and is given as

$$y_i = \phi(x_i; W_\ell) = T_{L+1} \circ \sigma \circ T_L \circ \cdots \circ \sigma \circ T_1(x_i) \quad (1)$$

where  $T_\ell(x) = W_\ell x$ ,  $\circ$  denotes composition of functions, and  $\sigma$  is an activation function  $\sigma(x) = \tanh(x)$ . Note that  $\phi(x) \in C^\infty$ .

Spectral normalization (SN) [6] is a technique to overcome the instability of neural network training by constraining (1) to be globally Lipschitz, i.e.,  $\exists L_{mn} \geq 0$  s.t.  $\|\phi(x) - \phi(x')\| \leq L_{mn}\|x - x'\|$ ,  $\forall x, x'$ , which is shown to be useful in nonlinear control designs [20]. SN normalizes the weight matrices  $W_\ell$  as  $W_\ell = (C_{mn}\Omega_\ell)/\|\Omega_\ell\|$  with  $C_{mn} \geq 0$  being a given constant, and trains a network with respect to  $\Omega_\ell$ . Since this results in  $\|\phi(x) - \phi(x')\| \leq C_{mn}^{L+1}\|x - x'\|$  [6], setting  $C_{mn} = L_{mn}^{1/(L+1)}$  guarantees Lipschitz continuity of  $\phi(x)$ . In Section III-B, we propose one way to use SN for building a neural network that guarantees the Lipschitz assumption on  $M_{x_i}$  in Theorem 1.

## B. Stochastic Contraction Analysis for Incremental Stability

Consider the following nonlinear system with stochastic perturbation given by the Itô stochastic differential equation:

$$dx = f(x, t)dt + G(x, t)dW(t), \quad x(0) = x_0 \quad (2)$$

where  $t \in \mathbb{R}_{\geq 0}$ ,  $x : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ ,  $f : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$ ,  $G : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times d}$ ,  $W(t)$  is a  $d$ -dimensional Wiener process, and  $x_0$  is a random variable independent of  $W(t)$  [21]. We assume that 1)  $\exists L_1 > 0$  s.t.  $\|f(x_1, t) - f(x_2, t)\| + \|G(x_1, t) - G(x_2, t)\|_F \leq L_1\|x_1 - x_2\|$ ,  $\forall t \in \mathbb{R}_{\geq 0}$  and  $\forall x_1, x_2 \in \mathbb{R}^n$ , and 2)  $\exists L_2 > 0$ , s.t.  $\|f(x_1, t)\|^2 + \|G(x_1, t)\|_F^2 \leq L_2(1 + \|x_1\|^2)$ ,  $\forall t \in \mathbb{R}_{\geq 0}$  and  $\forall x_1 \in \mathbb{R}^n$  for the sake of existence and uniqueness of the solution to (2).

Theorem 1 analyzes stochastic incremental stability of two trajectories of (2),  $x_1$  and  $x_2$ . In Section IV, we use Theorem 1 to find a contraction metric  $M(x, t)$  for given  $\alpha$ ,  $\varepsilon$ , and  $L_m$ , where  $\alpha$  is a contraction rate,  $\varepsilon$  is a parameter for disturbance attenuation, and  $L_m$  is the Lipschitz constant of  $M_{x_i}$ . Note that  $\varepsilon$  and  $L_m$  are introduced for the sake of stochastic contraction and were not present in the deterministic case [2]. Section IV-B2 delineates how we select them in practice.

*Theorem 1:* Suppose  $\exists g_1, g_2 \in [0, \infty)$  s.t.  $\|G(x_1, t)\|_F \leq g_1$  and  $\|G(x_2, t)\|_F \leq g_2$ ,  $\forall x, t$ . Suppose also that  $\exists M(x, t) \succ 0$  s.t.  $M_{x_i}$ ,  $\forall x_i$  is Lipschitz with respect to the state  $x$ , i.e.,  $\|M_{x_i}(x, t) - M_{x_i}(x', t)\| \leq L_m\|x - x'\|$ ,  $\forall x, x', t$  with  $L_m \geq 0$ . If  $M(x, t) \succ 0$  and  $\alpha, \varepsilon, \underline{\omega}, \bar{\omega} \in (0, \infty)$  are given by

$$\dot{M}(x, t) + 2\text{sym}(M(x, t)f_x(x, t)) + \alpha_g I \leq -2\alpha M(x, t) \quad (3)$$

$$\bar{\omega}^{-1} I \leq M(x, t) \leq \underline{\omega}^{-1} I, \quad \forall x, t \quad (4)$$

where  $\alpha_g = L_m(g_1^2 + g_2^2)(\varepsilon + 1/2)$ , then the mean squared distance between  $x_1$  and  $x_2$  is bounded as follows:

$$E[\|x_1 - x_2\|^2] \leq \frac{C}{2\alpha} \frac{\bar{\omega}}{\underline{\omega}} + \bar{\omega} E[V(x(0), \delta x(0), 0)] e^{-2\alpha t}. \quad (5)$$

where  $V(x, \delta x, t) = \delta x^T M(x, t) \delta x$  and  $C = (g_1^2 + g_2^2)(2/\varepsilon + 1)$ .

*Proof:* Let us first derive the bounds of  $M_{x_i}$  and  $M_{x_i x_j}$ . Since  $M_{x_i}$ ,  $\forall x_i$  is Lipschitz, we have  $\|M_{x_i x_j}\| \leq L_m$ ,  $\forall i, j$  by definition. For  $h \geq 0$  and a unit vector  $e_i$  with 1 in its  $i$ th element, the Taylor's theorem suggests  $\exists \xi_{\pm}, \xi_{\pm} \in \mathbb{R}^n$  s.t.

$$M(x \pm h e_i, t) = M(x, t) \pm M_{x_i}(x, t)h + M_{x_i x_i}(\xi_{\pm}, t)h^2/2. \quad (6)$$

This implies that  $\|M_{x_i}\|$  is bounded as  $\|M_{x_i}\| \leq h^{-1}\underline{\omega}^{-1} + L_m h/2 \leq \sqrt{2L_m\underline{\omega}^{-1}}$ , where  $h = \sqrt{2/(L_m\underline{\omega})}$  is substituted to obtain the last inequality. Next, let  $\mathcal{L}$  be the infinitesimal differential generator [8]. Computing  $\mathcal{L}V$  using these bounds as in [8] yields

$$\begin{aligned} \mathcal{L}V &\leq \delta x^T (\dot{M} + 2\text{sym}(Mf_x))\delta x \\ &\quad + (g_1^2 + g_2^2)(L_m\|\delta x\|^2/2 + 2\sqrt{2L_m\underline{\omega}^{-1}}\|\delta x\| + \underline{\omega}^{-1}) \\ &\leq \delta x^T (\dot{M} + 2\text{sym}(Mf_x) + \alpha_g I)\delta x + C\underline{\omega}^{-1} \end{aligned} \quad (7)$$

where the relation  $2ab \leq \varepsilon^{-1}a^2 + \varepsilon b^2$ , which holds for any  $a, b \in \mathbb{R}$  and  $\varepsilon > 0$ , is used with  $a = \sqrt{2/\underline{\omega}}$  and  $b = \sqrt{L_m}\|\delta x\|$  to get the second inequality. This reduces to  $\mathcal{L}V \leq -2\alpha V + C\underline{\omega}^{-1}$  under the condition (3). The result (5) follows as in the proof of [8, Th. 1]. ■

*Remark 1:* Note that there is a trade-off in using large  $\varepsilon$  in Theorem 1, as it yields small  $C$  to decrease the steady-state error in (5), but renders the constraint (3) tighter.

Lemma 1 is used to convexify the cost function in Section IV.

*Lemma 1:* The inequalities (3) and (4) are equivalent to

$$\begin{bmatrix} -\dot{\bar{W}} + 2\text{sym}(f_x(x, t)\bar{W}) + 2\alpha\bar{W} & \bar{W} \\ \bar{W} & -\frac{\nu}{\alpha_g}I \end{bmatrix} \leq 0 \quad (8)$$

$$I \leq \bar{W} \leq \chi I, \quad \forall x, t \quad (9)$$

where  $\nu = 1/\underline{\omega}$ ,  $\chi = \bar{\omega}/\underline{\omega}$ , and  $\bar{W} = \nu W = \nu M^{-1}$ .

*Proof:* Multiplying both sides of (3) by  $W > 0$  and then by  $\nu > 0$  preserves matrix definiteness [22, p. 114]. This operation with Schur's complement lemma [22, p. 28] yield (8). The rest follows the proof of [2, Lemma 1]. ■

*Remark 2:* The variable conversion in Lemma 1 is necessary to get a convex cost function (28) from the non-convex cost (5) as  $t \rightarrow \infty$ . In Section IV, we use it to derive a semi-definite program in terms of  $\nu$ ,  $\chi$ , and  $\bar{W}$  for finding a contraction metric computationally efficiently [23]. We show in Proposition 2 that this is equivalent to the non-convex problem of minimizing (5) as  $t \rightarrow \infty$ , subject to (3) and (4) in terms of the original decision variables  $\bar{\omega}$ ,  $\underline{\omega}$ , and  $M$  [8].

Finally, Lemma 2 introduces the generalized SDC form of dynamical systems to be exploited also in Section IV.

*Lemma 2:* Suppose that  $f(x, t)$  and  $B(x, t)$  are continuously differentiable. Then  $\exists A(x, x_d, t)$  s.t.  $A(x, x_d, t)(x - x_d) = f(x, t) + B(x, t)u_d(x_d, t) - f(x_d, t) - B(x_d, t)u_d(x_d, t)$ ,  $\forall x, x_d, u_d, t$ , and one such  $A$  is given as follows:

$$A(x, x_d, t) = \int_0^1 \frac{\partial \bar{f}}{\partial x}(cx + (1-c)x_d, t)dc \quad (10)$$

where  $\bar{f}(q, t) = f(q, t) + B(q, t)u_d(x_d, t)$ . We call  $A$  an SDC form when it is constructed to satisfy controllability and observability conditions (see Theorem 2 and Corollary 1).

*Proof:* This follows from the integral relation given as  $\int_0^1 (d\bar{f}(cx + (1-c)x_d, t)/dc)dc = \bar{f}(x, t) - \bar{f}(x_d, t)$ . ■

### III. NEURAL STOCHASTIC CONTRACTION METRICS

This section elucidates how to construct an NSCM using state samples  $S = \{x_i\}_{i=1}^N$  and stochastic contraction metrics given by Theorem 1. This is analogous to the NCM [2], which gives an optimal contraction metric for nonlinear systems with bounded disturbances, but the NSCM explicitly accounts for unbounded stochastic disturbances. For simplicity, we denote the metric both for feedback control and estimation as  $M$  with  $\underline{m}I \leq M \leq \bar{m}I$ . Note that we model  $W = M^{-1}$  for estimation.

### A. Data Pre-Processing

Since  $M > 0$ , where  $M$  is a contraction metric for control or estimation, it has a unique upper triangular matrix  $Y \in \mathbb{R}^{n \times n}$  with positive diagonal entries s.t.  $M = Y^T Y$  [24, p. 441]. We use the nonzero entries of  $Y$ , denoted as  $\theta(x, t) \in \mathbb{R}^{n(n+1)/2}$ , for  $y_i$  of (1) to reduce its output dimension [2].

### B. Lipschitz Condition and Spectral Normalization (SN)

We utilize SN in Section II-A to guarantee the Lipschitz condition of Theorem 1 or Proposition 2 in Section IV.

*Proposition 1:* Let  $\vartheta(x; W_{sn})$  be a neural network (1) to model  $\theta(x, t)$  in Section III-A, and  $N_{\text{units}}$  be the number of neurons in its last layer. Also, let  $W_{sn} = \{W_\ell\}_{\ell=1}^{L+1}$ , where  $W_\ell = (\Omega_\ell/\|\Omega_\ell\|)C_{nn}$  for  $1 \leq \ell \leq L$ , and  $W_\ell = \sqrt{\bar{m}}(\Omega_\ell/\|\Omega_\ell\|)/\sqrt{N_{\text{units}}}$  for  $\ell = L+1$ . If  $\exists C_{nn}, L_m > 0$  s.t.

$$\begin{aligned} 2\|\vartheta_{x_i}(x; W_{sn})\| \|\vartheta_{x_j}(x; W_{sn})\| \\ + 2\|\vartheta(x; W_{sn})\| \|\vartheta_{x_i x_j}(x; W_{sn})\| \leq L_m, \quad \forall i, j, x, \Omega \end{aligned} \quad (11)$$

then we have  $\|\mathcal{M}\| \leq \bar{m}$  and  $\|\mathcal{M}_{x_i x_j}\| \leq L_m$ ,  $\forall x_i, x_j$ , where  $\mathcal{M}$  is the neural network model for the contraction metric  $M(x, t)$ . The latter inequality implies  $\mathcal{M}_{x_i}$ ,  $\forall i$  is indeed Lipschitz continuous with 2-norm Lipschitz constant  $L_m$ .

*Proof:* Let  $\mathcal{Y}$  be the neural net model of  $Y$  in Section III-A. By definition of  $M = Y^T Y$  and  $\theta$ , where  $M$  is the contraction metric, we have  $\|\mathcal{M}\| \leq \|\mathcal{Y}\|^2 \leq \|\mathcal{Y}\|_F^2 = \|\vartheta\|^2$ . Thus, the relation  $\|\vartheta(x; W_{sn})\| \leq \sqrt{N_{\text{units}}}\|W_{L+1}\|$  yields  $\|\mathcal{M}\| \leq \bar{m}$  for  $W_{L+1} = \sqrt{\bar{m}}(\Omega_{L+1}/\|\Omega_{L+1}\|)/\sqrt{N_{\text{units}}}$ . Also, differentiating  $\mathcal{M}$  twice yields  $\|\mathcal{M}_{x_i x_j}\|/2 \leq \|\mathcal{Y}_{x_i}\| \|\mathcal{Y}_{x_j}\| + \|\mathcal{Y}\| \|\mathcal{Y}_{x_i x_j}\| \leq \|\vartheta_{x_i}\| \|\vartheta_{x_j}\| + \|\vartheta\| \|\vartheta_{x_i x_j}\|$ , where the second inequality is due to  $\|\mathcal{Y}\| \leq \|\mathcal{Y}\|_F = \|\vartheta\|$ . Substituting  $W_{sn}$  gives (11). ■

*Example 1:* To see how Proposition 1 works, let us consider a scalar input/output neural net with one neuron at each layer in (1). Since we have  $\|\vartheta(x; W_{sn})\| \leq \|W_{L+1}\|$ ,  $\mathcal{M} \leq \bar{m}I$  is indeed guaranteed by  $\|W_{L+1}\| = \sqrt{\bar{m}}$ . Also, we can get the bounds as  $\|\vartheta_{x_i}(x; W_{sn})\| \leq \sqrt{\bar{m}}C_{nn}^L$  and  $\|\vartheta_{xx}(x; W_{sn})\| \leq \|W_{L+1}\|C_{nn}^L(\sum_{\ell=1}^L C_{nn}^\ell) = \sqrt{\bar{m}}C_{nn}^{L+1}(C_{nn}^L - 1)/(C_{nn} - 1)$  using SN. Thus, (11) can be solved for  $C_{nn}$  by standard nonlinear equation solvers, treating  $\bar{m}$  and  $L_m$  as given constants.

*Remark 3:* For non-autonomous systems, we can treat  $t$  or time-varying parameters  $p(t)$  as another input to the neural network (1) by sampling them in a given parameter range of interest. For example, we could use  $p = [x_d, u_d]^T$  for systems with a target trajectory. This also allows us to use adaptive control techniques [25], [26] to update an estimate of  $p$ .

## IV. MCV-STEM SAMPLING OF CONTRACTION METRICS

We introduce the modified ConVex optimization-based Steady-state Tracking Error Minimization (mCV-STEM) method, an improved version of CV-STEM [8] for sampling the metrics which minimize an upper bound of the steady-state mean squared tracking error via convex optimization.

*Remark 4:* Due to its contraction-based formulation, combination properties [14] also apply to the NSCM framework. For example, contraction is preserved through hierarchical combination of estimation and control (i.e., output feedback control), or through time-delayed communications [15].

### A. Stability of Generalized SDC Control and Estimation

We utilize the general SDC parametrization with a target trajectory (10), which captures nonlinearity through  $A(x, x_d, t)$



or through multiple non-unique  $A_i$  [5], resulting in global exponential stability if the pair  $(A, B)$  of (12) is uniformly controllable [5], [7]. Note that  $x_d$  and  $u_d$  can be regarded as extra inputs to the NSCM as in Remark 3, but we could use Corollary 2 as a simpler formulation which guarantees local exponential stability without using a target trajectory. Further extension to control contraction metrics, which use differential state feedback  $\delta u = K(x, t)\delta x$  [9]–[13], could be considered for sampling the metric with global reference-independent stability guarantees, achieving greater generality at the cost of added computation. Similarly, while we construct an estimator with global stability guarantees using the SDC form as in (22), a more general formulation could utilize geodesics distances between trajectories [4]. We remark that these trade-offs would also hold for deterministic control and estimation design via NCMs [2].

**1) Generalized SDC Control:** Consider the following system with a controller  $u \in \mathbb{R}^m$  and perturbation  $\mathcal{W}(t)$ :

$$dx = (f(x, t) + B(x, t)u)dt + G_e(x, t)d\mathcal{W}(t) \quad (12)$$

$$dx_d = (f(x_d, t) + B(x_d, t)u_d(x_d, t))dt \quad (13)$$

where  $B : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times m}$ ,  $G_e : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times d}$ ,  $\mathcal{W}(t)$  is a  $d$ -dimensional Wiener process, and  $x_d : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^n$  and  $u_d : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$  denote the target trajectory.

**Theorem 2:** Suppose  $\exists g_c \in [0, \infty)$  s.t.  $\|G_e(x, t)\|_F \leq g_c$ ,  $\forall x, t$ , and  $\exists M(x, x_d, t) > 0$  s.t.  $M_{x_i}$  and  $M_{x_{d,i}}$ ,  $\forall x_i, x_{d,i}$  are Lipschitz with respect to its state with 2-norm Lipschitz constant  $L_m$ . Let  $u$  be designed as

$$u = u_d(x_d, t) - B(x, t)^T M(x, x_d, t)(x - x_d) \quad (14)$$

$$\dot{M} + 2\text{sym}(MA) - 2MBB^T M + \alpha_{gc}I \leq -2\alpha M \quad (15)$$

$$\bar{\omega}^{-1}I \leq M(x, x_d, t) \leq \underline{\omega}^{-1}I, \quad \forall x, t \quad (16)$$

where  $\alpha > 0$ ,  $\alpha_{gc} = L_m g_c^2(\varepsilon + 1/2)$ ,  $\varepsilon > 0$ , and  $A$  is given by (10) in Lemma 2. If the pair  $(A, B)$  is uniformly controllable, we have the following bound for the systems (12) and (13):

$$E[\|x - x_d\|^2] \leq \frac{C_c}{2\alpha} \chi + \bar{\omega} E[V(x(0), x_d(0), \delta q(0), 0)]e^{-2\alpha t} \quad (17)$$

where  $V(x, x_d, \delta q, t) = \delta q^T M(x, x_d, t)\delta q$ ,  $C_c = g_c^2(2/\varepsilon + 1)$ ,  $v = 1/\underline{\omega}$ ,  $\chi = \bar{\omega}/\underline{\omega}$ , and  $q$  is the state of the differential system with its particular solutions,  $q = x$  and  $q = x_d$ . Further, (15) and (16) are equivalent to the following constraints in terms of  $v$ ,  $\chi$ , and  $\bar{W} = vW = vM^{-1}$ :

$$\begin{bmatrix} -\dot{\bar{W}} + 2\text{sym}(A\bar{W}) - 2vBB^T + 2\alpha\bar{W} & \bar{W} \\ \bar{W} & -\frac{v}{\alpha_{gc}}I \end{bmatrix} \leq 0 \quad (18)$$

$$I \leq \bar{W} \leq \chi I, \quad \forall x, t. \quad (19)$$

where the arguments are omitted for notational simplicity.

**Proof:** Using the SDC parameterization (10) given in Lemma 2, (12) can be written as  $dx = (\hat{f}(x_d, t) + (A(x, x_d, t) - B(x, t)B(x, t)^T M(x, x_d, t))(x - x_d))dt + G_e(x, t)d\mathcal{W}$ . This results in the following differential system,  $dq = (\hat{f}(x_d, t) + (A(x, x_d, t) - B(x, t)B(x, t)^T M(x, x_d, t))(q - x_d))dt + G(q, t)d\mathcal{W}$ , where  $G(q, t)$  is defined as  $G(q = x, t) = G_e(x, t)$  and  $G(q = x_d, t) = 0$ . Note that it has  $q = x, x_d$  as its particular solutions. Since  $\hat{f}_x$ ,  $g_1$ , and  $g_2$  in Theorem 1 can be viewed as  $A(x, x_d, t) - B(x, t)B(x, t)^T M(x, x_d, t)$ ,  $g_c$ , and 0, respectively, applying its results for  $V = \delta q^T M(x, x_d, t)\delta q$  gives (17) as in (5). The constraints (18) and (19) follow from the application of Lemma 1 to (15) and (16). ■

**Remark 5:** For input non-affine nonlinear systems, we can find  $f(x, u) - f(x_d, u_d) = A(x, u, t)(x - x_d) + B(x, u, t)(u - u_d)$

by Lemma 2 and use it in Theorem 2, although (14) has to be solved implicitly as  $B$  depends on  $u$  in this case [12], [13].

**2) Generalized SDC Estimation:** Consider the following system and a measurement  $y(t)$  with perturbation  $\mathcal{W}_{1,2}(t)$ :

$$dx = f(x, t)dt + G_e(x, t)d\mathcal{W}_1(t) \quad (20)$$

$$ydt = h(x, t)dt + D(x, t)d\mathcal{W}_2(t) \quad (21)$$

where  $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$ ,  $G_e : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{n \times d_1}$ ,  $D : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^{m \times d_2}$ , and  $\mathcal{W}_{1,2}(t)$  are two independent Wiener processes. We have an analogous result to Theorem 2.

**Corollary 1:** Suppose  $\exists g_e, \bar{d} \in [0, \infty)$  s.t.  $\|G_e(x, t)\|_F \leq g_e$  and  $\|D(x, t)\|_F \leq \bar{d}$ ,  $\forall x, t$ . Suppose also that  $\exists W(\hat{x}, t) = M(\hat{x}, t)^{-1} > 0$  s.t.  $W_{x_i}$ ,  $\forall x_i$  is Lipschitz with respect to its state with 2-norm Lipschitz constant  $L_m$ . Let  $v = 1/\underline{\omega}$  and  $x$  be estimated as

$$d\hat{x} = f(\hat{x}, t)dt + M(\hat{x}, t)C_L(\hat{x}, t)^T(y - h(\hat{x}, t))dt \quad (22)$$

$$\dot{W} + 2\text{sym}(WA - C_L^T C) + \alpha_{ge}I \leq -2\alpha W \quad (23)$$

$$\underline{\omega}I \leq W(\hat{x}, t) \leq \bar{\omega}I, \quad 0 < v \leq \sqrt[3]{v_c}, \quad \forall x, \hat{x}, t \quad (24)$$

where  $\alpha, v_c, \varepsilon > 0$ ,  $\alpha_{ge} = \alpha_{e1} + v_c \underline{\omega} \alpha_{e2}$ ,  $\alpha_{e1} = L_m g_e^2(\varepsilon + 1/2)$ , and  $\alpha_{e2} = L_m \bar{c}^2 \bar{d}^2(\varepsilon + 1/2)$ . Also,  $A(x, \hat{x}, t)$  and  $C(x, \hat{x}, t)$  are given by (10) of Lemma 2 with  $(f, x, x_d, u_d)$  replaced by  $(f, \hat{x}, x, 0)$  and  $(h, \hat{x}, x, 0)$ , respectively, and  $C_L(\hat{x}, t) = \hat{C}(\hat{x}, \hat{x}, t)$ . If  $(A, C)$  is uniformly observable and  $\|C(x, \hat{x}, t)\| \leq \bar{c}$ ,  $\forall x, \hat{x}, t$ , then we have the following bound:

$$E[\|x - \hat{x}\|^2] \leq \frac{C_e}{2\alpha} \chi + \frac{1}{\underline{\omega}} E[V(x(0), \delta q(0), 0)]e^{-2\alpha t} \quad (25)$$

where  $V(\hat{x}, \delta q, t) = \delta q^T W(\hat{x}, t)\delta q$ ,  $C_e = C_{e1} + C_{e2}v^2$ ,  $C_{e1} = g_e^2(2/\varepsilon + 1)$ ,  $C_{e2} = \bar{c}^2 \bar{d}^2(2/\varepsilon + 1)$ ,  $\chi = \bar{\omega}/\underline{\omega}$ , and  $q$  is the state of the differential system with its particular solutions  $q = \hat{x}, x$ . Further, (23) and (24) are equivalent to the following constraints in terms of  $v$ ,  $v_c$ ,  $\chi$ , and  $\bar{W} = vW$ :

$$\dot{\bar{W}} + 2\text{sym}(\bar{W}A - vC_L^T C) + v\alpha_{e1}I + v_c\alpha_{e2}I \leq -2\alpha \bar{W} \quad (26)$$

$$I \leq \bar{W} \leq \chi I, \quad 0 < v \leq \sqrt[3]{v_c}, \quad \forall x, \hat{x}, t \quad (27)$$

where the arguments are omitted for notational simplicity.

**Proof:** The differential system of (20) and (22) is given as  $dq = f(x, t) + (A(x, \hat{x}, t) - M(\hat{x}, t)C_L(\hat{x}, t)^T C(x, \hat{x}, t))(q - x)dt + G(q, t)d\mathcal{W}$ , where  $G(q, t)$  is defined as  $G(q = x, t) = G_e(x, t)$  and  $G(q = \hat{x}, t) = M(\hat{x}, t)C_L(\hat{x}, t)^T D(x, t)$ . Viewing  $V$ ,  $g_1$ , and  $g_2$  in Theorem 1 as  $V = \delta q^T W(\hat{x}, t)\delta q$ ,  $g_1 = g_e$ , and  $g_2 = \bar{c}\bar{d}/\underline{\omega}$ , (25) – (27) follow as in the proof of Theorem 2 due to  $v^3 = \underline{\omega}^{-3} \leq v_c$  and the contraction condition (23). ■

Note that (15) and (23) depend on their target trajectory, i.e.,  $x_d$  for control and  $x$  for estimation. We can treat them as time-varying parameters  $p(t)$  in a given space during the mCV-STEM sampling as in Remark 3. Alternatively, we could use the following to avoid this complication.

**Corollary 2:** Using predefined trajectories (e.g.,  $(x_d, u_d) = (0, 0)$  for control or  $x = 0$  for estimation) in Theorem 2 or Corollary 1 leads to local exponential stability of (12) or (22).

**Proof:** This follows as in the proof of Theorem 2 [2]. ■

## B. mCV-STEM Formulation

The following proposition summarizes the mCV-STEM.

**Proposition 2:** The optimal contraction metric  $M = W^{-1}$  that minimizes the upper bound of the steady-state mean squared distance ((17) of Theorem 2 or (25) of Cor. 1 with  $t \rightarrow \infty$ ) of stochastically perturbed system trajectories is found by the following convex optimization problem:

**Algorithm 1:** NSCM Algorithm

**Inputs :** States & parameters:  $S = \{x_i\}_{i=1}^N$  or  $\{\hat{x}_i\}_{i=1}^N$   
 &  $T = \{p_i\}_{i=1}^M$  (e.g.,  $p = t$ ,  $[x_d, u_d]^T$ , or  $x$ )

**Outputs:** NSCM and  $J_{CV}^*$  in (28)

## 1. Sampling of Optimal Contraction Metrics

Find  $L_m$  in Theorem 1 using a deterministic NCM [2]

**for**  $(\alpha, \varepsilon) \in A_{LS}$  ( $A_{LS}$  is a search set) **do**

    Solve (28) of Proposition 2 using  $x, p$  or  $\hat{x}, p$  in  $S$   
 &  $T$

    Save the optimizer  $(v, \chi, \{\bar{W}_i\}_{i=1}^N)$  and optimal  
 value  $J(\alpha, \varepsilon) = c_1\chi + c_2v + c_3P(v, v_c, \chi, \bar{W})$

Find  $(\alpha^*, \varepsilon^*) = \arg \min_{(\alpha, \varepsilon) \in A_{LS}} J$  and  $J_{CV}^* = J(\alpha^*, \varepsilon^*)$

Obtain  $(v(\alpha^*, \varepsilon^*), \chi(\alpha^*, \varepsilon^*), \{\bar{W}_i(\alpha^*, \varepsilon^*)\}_{i=1}^N)$

## 2. Spectrally-Normalized Neural Network Training

Pre-process data as in Section III-A

Split data into a train set  $S_{\text{train}}$  and test set  $S_{\text{test}}$

**for** epoch  $\leftarrow 1$  **to**  $N_{\text{epochs}}$  **do**

**for**  $s \in S_{\text{train}}$  **do**

        Train a neural network using SGD with the  
 Lipschitz condition as in Proposition 1

    Compute the test error for data in  $S_{\text{test}}$

**if** test error is small enough **then**

**break**

$$J_{CV}^* = \min_{v>0, v_c>0, \chi \in \mathbb{R}, \bar{W}>0} c_1\chi + c_2v + c_3P(v, v_c, \chi, \bar{W}) \quad (28)$$

s.t. (18) & (19) for control, (26) & (27) for estimation

where  $c_1, c_2, c_3 \in [0, \infty)$  and  $P$  is an additional performance-based convex cost (see Section IV-B1). The weight of  $v, c_2$ , can either be viewed as a penalty on the 2-norm of feedback gains or an indicator of how much we trust the measurement  $y(t)$ . Note that  $\alpha, \varepsilon$ , and  $L_m$  are assumed to be given in (28) (see Section IV-B2 for how to handle  $\bar{W}$  preserving convexity).

*Proof:* For control (17), using  $c_1 = C_c/(2\alpha)$  and  $c_2 = c_3 = 0$  gives (28). We can set  $c_2 > 0$  to penalize excessively large  $\|u\|$  through  $v \geq \sup_{x,t} \|M(x, x_d, t)\|$ . For estimation, since we have  $v > 0$  and  $1 \leq \chi \leq \chi^3$ , (25) as  $t \rightarrow \infty$  can be bounded as

$$\frac{C_{e1}\chi + C_{e2}\chi v^2}{2\alpha} \leq \frac{1}{3\sqrt{3}C_{e1}} \left( \frac{\sqrt{3}C_{e1}}{\sqrt[3]{2\alpha}} \chi + \frac{\sqrt{C_{e2}}}{\sqrt[3]{2\alpha}} v \right)^3. \quad (29)$$

Minimizing the right-hand side of (29) gives (28) with  $c_1 = \sqrt{3}C_{e1}/\sqrt[3]{2\alpha}$ ,  $c_2 = \sqrt{C_{e2}}/\sqrt[3]{2\alpha}$ , and  $c_3 = 0$ . Finally, since  $\bar{d} = 0$  in (21) means  $C_{e2} = 0$  and no noise acts on  $y$ ,  $c_2$  also indicates how much we trust the measurement. ■

1) *Choice of  $P(v, v_c, \chi, \bar{W})$ :* Selecting  $c_3 = 0$  in Proposition 2 yields an affine objective function that allows for a straightforward interpretation of its weights. Users could also select  $c_3 > 0$  with other performance-based cost functions  $P(v, v_c, \chi, \bar{W})$  in (28) as long as they are convex. For example, an objective function  $\sum_{x_i \in S} \|u\|^2 = \sum_{x_i \in S} \| -B(x_i, t)^T M(x_i, t) x_i \|^2 \leq \sum_{x_i \in S} \|B(x_i, t)\|^2 \|x_i\|^2 v^2$ , where  $S$  is the state space of interest, gives an optimal contraction metric which minimizes the upper bound of its control effort.

2) *Additional Parameters and  $\bar{W}$ :* We assumed  $\alpha, \varepsilon$ , and  $L_m$  are given in Proposition 2. For  $\alpha$  and  $\varepsilon$ , we perform a line search to find their optimal values as will be demonstrated in Section V. For  $L_m$ , we guess it by a deterministic NCM [2] and guarantee the Lipschitz condition by SN as explained in

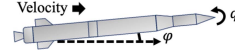


Fig. 2. Rocket model (angle of attack  $\varphi$ , pitch rate  $q$ ).

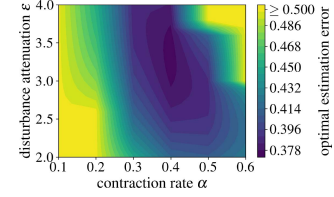


Fig. 3. Optimal steady-state estimation error as a function of  $\alpha$  and  $\varepsilon$ .

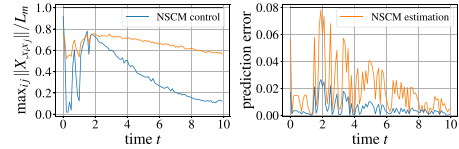


Fig. 4. NSCM spectral normalization and prediction error.

TABLE I  
NSCM CONTROL AND ESTIMATION PARAMETERS

	$\alpha$	$\varepsilon$	$L_m$	steady-state upper bound
estimation	0.40	3.30	0.50	0.41
control	0.10	1.00	10.00	0.58

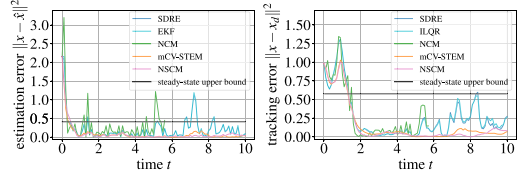


Fig. 5. Rocket state estimation and tracking error ( $x = [\varphi, q]^T$ ).

Section III-B. Also, (28) can be solved as a finite-dimensional problem by using backward difference approximation on  $\bar{W}$ , where we can then use  $-\bar{W} \leq -I$  to obtain a sufficient condition of its constraints, or solve it along pre-computed trajectories  $\{x(t_i)\}_{i=0}^M$  [2], [27]. The pseudocode to obtain the NSCM depicted in Fig. 1 is given in Algorithm 1.

## V. NUMERICAL IMPLEMENTATION EXAMPLE

We demonstrate the NSCM on a rocket autopilot problem (<https://github.com/astrohiro/nscm>). CVXPY [28] with the MOSEK solver [29] is used to solve convex optimization.

## A. Simulation Setup

We use the nonlinear rocket model in Fig. 2 [30], assuming  $q$  and specific normal force are available via rate gyros and accelerometers. We use  $G_c = (6.0e-2)I$ ,  $G_e = (3.0e-2)I$ , and  $D = (3.0e-2)I$  for perturbation in the NSCM construction. The Mach number is varied linearly in time from 2 to 4.

## B. NSCM Construction

We construct NSCMs by Algorithm 1. For estimation, we select the Lipschitz constant to be  $L_m = 0.50$  (see Section IV-B2). The optimal  $\alpha$  and  $\varepsilon$ ,  $\alpha^* = 0.40$  and  $\varepsilon^* = 3.30$ , are found by line search in Fig. 3. A neural net with 3 layers and 100 neurons is trained using  $N = 1000$  samples, where its SN constant is selected as  $C_{nn} = 0.85$  as

a result of Proposition 1. We use the same approach for the NSCM control and the resultant design parameters are given in Table I. Figure 4 implies that the NSCMs indeed satisfy the Lipschitz condition with its prediction error smaller than 0.08 thanks to SN.

## VI. DISCUSSION AND CONCLUDING REMARKS

We compare the NSCM with the SDRE [7], ILQR [16], [17], EKF, NCM [2], and mCV-STEM. As shown in Fig. 5, the steady-state errors of the NSCM and mCV-STEM are indeed smaller than its steady-state upper bounds (17) and (25) found by Proposition 2, while other controllers violate this condition. Also, the optimal contraction rate of the NCM for state estimation is much larger ( $\alpha = 6.1$ ) than the NSCM as it does not account for stochastic perturbation. This renders the NCM trajectory to diverge around  $t = 5.8$  in Fig. 5. The NSCM Lipschitz condition on the metric guaranteed by SN as in Fig. 4 allows us to circumvent this difficulty.

In conclusion, the NSCM is a novel way of using spectrally-normalized deep neural networks for real-time computation of learning-based nonlinear control and estimation policies, which are optimal and provably stable in the mean squared error sense even under stochastic disturbances. The reference-independent policies [4], [9]–[13] or the generalized SDC policies (14) and (22) introduced in this letter, which guarantee global exponential stability with respect to a target trajectory, could be used both in stochastic and deterministic frameworks including the NCM [2]. It is also noted that the combination properties of contraction theory in Remark 4 still holds for the deterministic NCM. An important future direction is to consider a model-free version of these techniques [31].

## ACKNOWLEDGMENT

This work was benefited from discussions with Nicholas Boffi and Quang-Cuong Pham. Code: <https://github.com/astrohiro/nscm>.

## REFERENCES

- [1] H. J. Kushner, *Stochastic Stability and Control*. New York, NY, USA: Academic, 1967. [Online]. Available: <https://www.elsevier.com/books/stochastic-stability-and-control/kushner/978-0-12-430150-4>
- [2] H. Tsukamoto and S.-J. Chung, “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 211–216, Jan. 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9115010>
- [3] Q. C. Pham, N. Tabareau, and J.-J. E. Slotine, “A contraction theory approach to stochastic incremental stability,” *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 816–820, Apr. 2009. [Online]. Available: <https://ieeexplore.ieee.org/document/4806161>
- [4] Q. C. Pham and J.-J. E. Slotine, (Apr. 2013). *Stochastic Contraction in Riemannian Metrics*. [Online]. Available: <https://arxiv.org/abs/1304.0340>
- [5] A. P. Dani, S.-J. Chung, and S. Hutchinson, “Observer design for stochastic nonlinear systems via contraction-based incremental stability,” *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 700–714, Mar. 2015. [Online]. Available: <https://ieeexplore.ieee.org/document/6899639>
- [6] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–26. [Online]. Available: <https://openreview.net/forum?id=B1QRgziT>
- [7] J. R. Cloutier, “State-dependent Riccati equation techniques: An overview,” in *Proc. Amer. Control Conf.*, vol. 2, 1997, pp. 932–936. [Online]. Available: <https://ieeexplore.ieee.org/document/609663>
- [8] H. Tsukamoto and S.-J. Chung, “Robust controller design for stochastic nonlinear systems via convex optimization,” *IEEE Trans. Autom. Control*, early access, Oct. 2021. [Online]. Available: <https://arxiv.org/abs/2006.04359>
- [9] I. R. Manchester and J.-J. E. Slotine, “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 3046–3053, Jun. 2017.
- [10] S. Singh, A. Majumdar, J.-J. E. Slotine, and M. Pavone, “Robust online motion planning via contraction theory and convex optimization,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 5883–5890. [Online]. Available: <https://ieeexplore.ieee.org/document/7989693>
- [11] S. Singh, V. Sindhvani, J.-J. E. Slotine, and M. Pavone, “Learning stabilizable dynamical systems via control contraction metrics,” in *Proc. Workshop Algorithmic Found. Robot.*, 2018. [Online]. Available: <https://arxiv.org/pdf/1808.00113.pdf>
- [12] R. Wang, R. Tóth, and I. R. Manchester, “A comparison of LPV gain scheduling and control contraction metrics for nonlinear control,” *IFAC PapersOnLine*, vol. 52, no. 28, pp. 44–49, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896319322463>
- [13] R. Wang, R. Tóth, and I. R. Manchester, *Virtual Control Contraction Metrics: Convex Nonlinear Feedback Design Via Behavioral Embedding*. Accessed: Mar. 2020. [Online]. Available: <https://arxiv.org/abs/2003.08513>
- [14] W. Lohmiller and J.-J. E. Slotine, “On contraction analysis for nonlinear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, 1998. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109898000193>
- [15] W. Wang and J.-J. E. Slotine, “Contraction analysis of time-delayed communications and group cooperation,” *IEEE Trans. Autom. Control*, vol. 51, no. 4, pp. 712–717, Apr. 2006. [Online]. Available: <https://ieeexplore.ieee.org/document/1618853?arnumber=1618853>
- [16] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems,” in *Proc. Int. Conf. Inf. Control Autom. Robot.*, 2004, pp. 222–229. [Online]. Available: <https://homes.cs.washington.edu/todorov/papers/LiICINCO04.pdf>
- [17] W. Li and E. Todorov, “An iterative optimal control and estimation design for nonlinear stochastic system,” in *Proc. 45th IEEE Conf. Decis. Control*, 2006, pp. 3242–3247. [Online]. Available: <https://ieeexplore.ieee.org/document/4177797>
- [18] S. M. Richards, F. Berkenkamp, and A. Krause, “The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems,” in *Proc. CoRL*, vol. 87, Oct. 2018, pp. 466–476. [Online]. Available: <https://arxiv.org/abs/1808.00924>
- [19] Y.-C. Chang, N. Roohi, and S. Gao, “Neural Lyapunov control,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3245–3254. [Online]. Available: <https://papers.nips.cc/paper/8587-neural-lyapunov-control>
- [20] G. Shi et al., “Neural lander: Stable drone landing control using learned dynamics,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2019, pp. 9784–9790. [Online]. Available: <https://ieeexplore.ieee.org/document/8794351>
- [21] L. Arnold, *Stochastic Differential Equations: Theory and Applications*. Hoboken, NJ, USA: Wiley, 1974. [Online]. Available: <https://epubs.siam.org/doi/10.1137/1018036?mobileUi=0>
- [22] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* (Studies in Applied Mathematics), vol. 15. Philadelphia, PA, USA: SIAM, Jun. 1994. [Online]. Available: <https://web.stanford.edu/boyd/lmibook/>
- [23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, Mar. 2004. [Online]. Available: <https://web.stanford.edu/boyd/cvxbook/>
- [24] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2012. [Online]. Available: <https://www.cambridge.org/core/books/matrix-analysis/9CF2CB491C9E97948B15FAD835EF9A8B>
- [25] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Upper Saddle River, NJ, USA: Pearson, 1991. [Online]. Available: <https://www.pearson.com/us/higher-education/program/Slotine-Applied-Nonlinear-Control/PGM175386.html>
- [26] B. T. Lopez and J.-J. E. Slotine, “Adaptive nonlinear control with contraction metrics,” *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 205–210, Jan. 2021. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9109296>
- [27] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: <https://dl.acm.org/doi/10.1162/neco.1997.9.8.1735>
- [28] S. Diamond and S. Boyd, “CVXPY: A Python-embedded modeling language for convex optimization,” *J. Mach. Learn. Res.*, vol. 17, p. 83, Apr. 2016. [Online]. Available: <https://www.cvxpy.org/>
- [29] *MOSEK Optimizer API for Python 9.0.105*, MOSEK ApS, Copenhagen, Denmark, 2020. [Online]. Available: <https://www.mosek.com/documentation/>
- [30] J. S. Shamma and J. R. Cloutier, “Gain-scheduled missile autopilot design using linear parameter varying transformations,” *J. Guid. Control Dyn.*, vol. 16, no. 2, pp. 256–263, 1993. [Online]. Available: <https://doi.org/10.2514/3.20997>
- [31] N. M. Boffi, S. Tu, N. Matni, J.-J. E. Slotine, and V. Sindhvani, “Learning stability certificates from data,” in *Proc. CoRL*, Nov. 2020. [Online]. Available: <https://arxiv.org/abs/2008.05952>